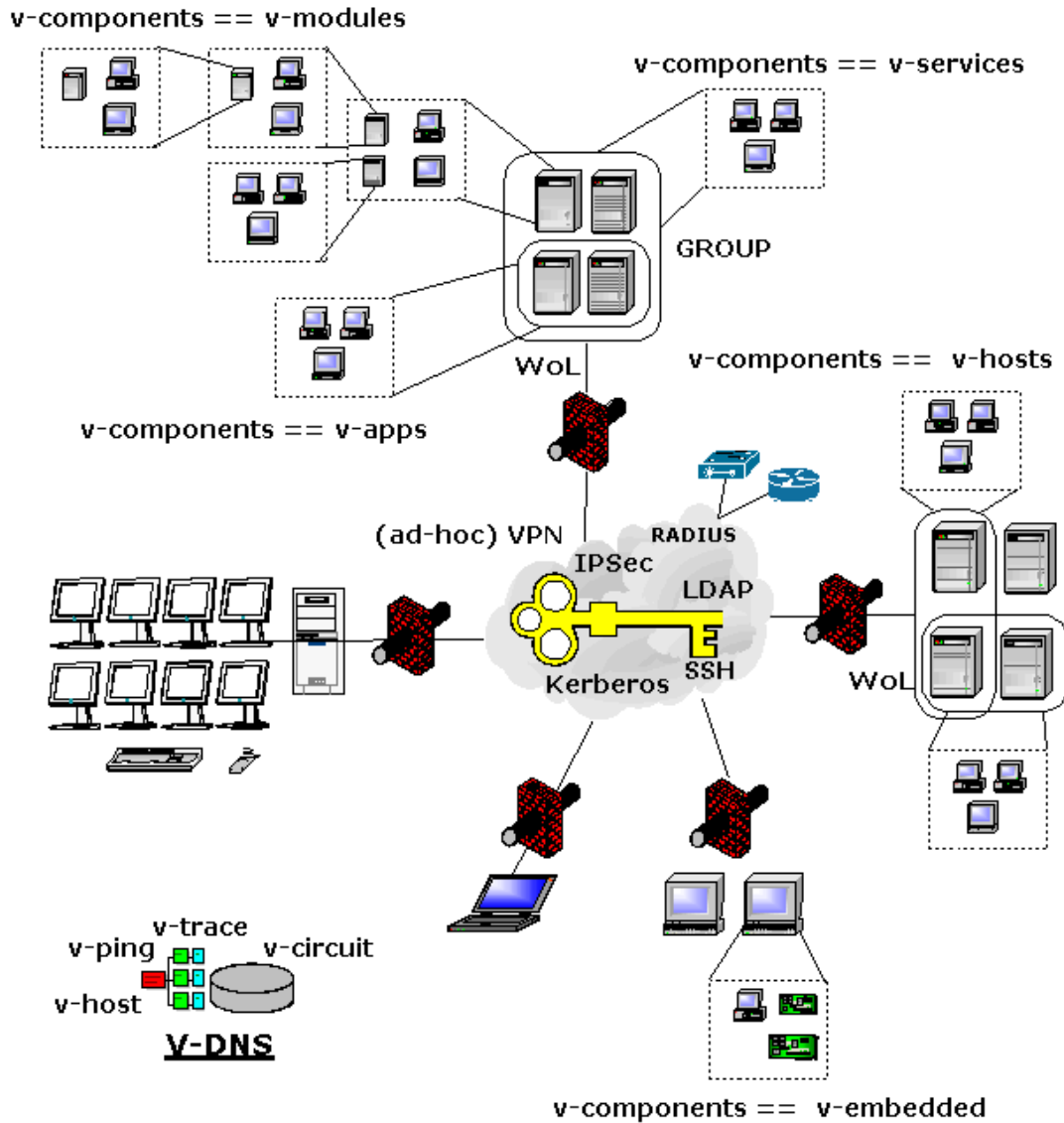


stacked-VMs == v-components



The UnifiedSessionsManager

QEMU Setup and Call Guide(draft-pre-release)

Extract from the User-Manual

Version:01.07.001b05a - 2009.01.26

Copyright 2008 by

Arno-Can Uestuensoez

Contents

1	Preface	11
1.1	History	11
1.2	Contact	12
1.3	Legal	12
2	Installation of QEMU	13
2.1	Install rpm	13
2.2	Install from Sources - Build and Install QEMU	13
2.3	Setup Networking for QEMU by VDE	14
2.3.1	Overview	14
2.3.2	VDE	16
2.3.3	QEMU	16
2.4	Setup Serial Console	18
	overview	18
3	Installation of Guests	21
3.1	PXE-Boot	21
3.2	Install Procedures	21
3.2.1	Debian-4.0_r3 by PXE - i386	21
3.2.2	Debian-4.0_r3 for ARM	21
3.2.3	OpenBSD-4.3+SerialConsole - by PXE	23
3.3	Installed Systems	24
4	Call Examples	25
4.1	CREATE a session	25
4.2	CANCEL a session	28
4.3	LIST sessions	28
4.4	ENUMERATE sessions	29
4.5	SHOW	29
4.6	INFO	29
5	Example Appliances	33
5.1	Qemu Test and Demo Examples	33
5.1.1	ARM	33
5.1.2	Coldfire	34

6 System Resources	35
6.1 TAP/TUN by VDE	35
6.1.1 VDE within Xen Dom0	36
6.1.2 VDE within Xen DomU	38
6.1.3 VDE within VMware	38
6.1.4 VDE within Native Unix	39
6.1.5 VDE Remote Configuration	41
6.1.6 VDE-Setup with Micro-VMSTACK	44
6.2 Mount an ISO-Image	45
Bibliography	47
Books	47
UNIX	47
Security	48
Networks	49
Embedded Systems	50
Online References	51
OSs	51
Hypervisors/Emulators	52
kvm	52
QEMU	52
SkyEye	52
VMware	53
Xen	53
Security	53
Specials	54
Dynagen/Dynamips	54
QEMU-Networking with VDE	54
PXE	54
Routing	54
Scratchbox	54
Serial-Console	54
Miscellaneous	54
UnifiedSessionsManager Versions	55
Sponsored OpenSource Projects	56
Commercial Support	56

List of Tables

3.1 Overview of Installed-QEMU-VMs	24
--	----

List of Figures

2.1	QEMU Interconnection-Interfaces	18
4.1	The "CONSOLE:EMACSAM" for a QEMU Session	27
4.2	The "CONSOLE:EMACSAM" and a "CONSOLE:VNC" for a QEMU Session	28
6.1	Virtual switch by vde_switch	36
6.2	QEMU NIC interconnection	37
6.3	QEMU NIC interconnection for Native UNIX	39

Licence - GPL3

Copyright (C) 2007,2008 Arno-Can Uestuensoez (UnifiedSessionsManager.org)

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

```
-----  
PROJECT           = Unified Sessions Manager  
-----  
CALLFULLNAME     = Commutate To Your Session  
CALLSHORTCUT     = ctys  
  
AUTHOR           = Arno-Can Uestuensoez - acue@UnifiedSessionsManager.org  
MAINTAINER       = Arno-Can Uestuensoez - acue_sf1@sourceforge.net  
VERSION          = 01_07_001b02  
DATE             = 2008.08.06  
  
COPYRIGHT        = Arno-Can Uestuensoez - acue@UnifiedSessionsManager.org  
LICENCE          = GPL3  
-----  
EXECUTING HOST   = ws2.soho  
-----
```


Chapter 1

Preface

This is just a draft extract from the User-Manual, in order to provide a shortcut for first installation.

Therefore some hyperlinks may be dangling or just reference a page referencing the user manual.

1.1 History

Version	Date	Author	Description
01.03.003.a01 [144]	2008.02.11	Arno-Can Uestuenseoz	Initial pre-release as embedded printable help
01.07.001.a01 [145]	2008.08.03	Arno-Can Uestuenseoz	First major update with numerous additions and partial review.
01.07.001.b02 [146]	2008.08.11	Arno-Can Uestuenseoz	Minor editorial updates. A lot of tests, some fixes.
01.07.001.b03 [147]	2008.08.12	Arno-Can Uestuenseoz	Minor editorial updates.
01.07.001.b04 [148]	2008.08.16	Arno-Can Uestuenseoz	Enhancement of documentation and Web-Site.
01.07.001.b05a [149]	2009.01.26	Arno-Can Uestuenseoz	Minor editorials.

01.03.003.a01/2008.02.11

The first basic set of features including stack-aware recursive cancel for QEMU, XEN, and VMW. This release contains almost the whole set of first-time implementations(Section ?? ‘??’ on page ??) with personal copyright and offered license under GPL3.

01.07.001.a01/2008.08.03

First major update with numerous additions and partial review. Additional HTML versions and subsets are generated. Still classified as preview quality, but may be stable enough for public production use, as personally applied.

01.07.001.b02/2008.08.11

The first official stable release for flat-VM functions. The stacked VMs are present in first version, where the canonical syntax is implemented. Some more user friendly addressing with enhanced capability-recognition and validation will follow.

1.2 Contact

Public maintenance:	acue@sfl_sourceforge.net
Administrative contact:	acue@UnifiedSessionsManager.org
Commercial Services:	i4p.com - Engineering Office Arno-Can Uestuensoez

1.3 Legal

All mentioned Red Hat products and their registered names are Trademarks of the Company Red Hat, Inc.

All mentioned SuSE products and their registered names are Trademarks of the Company Novell, Inc.

All mentioned VMware products and their registered names are Trademarks of the Company VMware, Inc.

All mentioned Google products and their registered names are Trademarks of the Google, Inc.

All mentioned Sun products and their registered names are Trademarks of the Sun Microsystems, Inc.

All mentioned Microsoft products and their registered names are Trademarks of the Company Microsoft, Inc.

All mentioned Intel products and their registered names are Trademarks of the Company Intel, Inc.

All mentioned AMD products and their registered names are Trademarks of the Company Advanced Micro Devices, Inc.

Xen is a trademark of XenSource Inc.

QEMU is a trademark of Fabrice Bellard.

i4p is a trademark of Ingenieurbuero Arno-Can Uestuensoez.

If some is forgotten, it will be added immediately.

Chapter 2

Installation of QEMU

2.1 Install rpm

The installation of the RPM package is quite forward. At least for Fedora the paths are unchanged from the original "configure" output.

2.2 Install from Sources - Build and Install QEMU

The QEMU installation itself is quite straight forward, the only point to be careful about is the requirement of a gcc3x compiler, gcc4 is not compatible. This is required for the configuration and compilation only, thus the installation could be performed on gcc4x.

The qemu component was not installed for now, because some VM stack tests with nested execution are required, and side-effects should have been excluded. Use the "-no-kqemu" flag for the qemu-call for version >0.9.1.

Previous of installation the build from the sources was executed. Therefore a 3x version of gcc is used as required. Due to the lack of a native install a VMware installation of SuSE-9.3 was used.

BUILD

On SuSE-9.3 running in VMware with gcc3.x:

1. Install gcc3x and SDL, un-tar "qemu-0.9.1.tar.gz"
2. call "configure"
3. call "make"
4. copy to CentOS, the paths has to be the same, due to "configure" resolution for "make install", so choose a common or network directory, such as "/tmp" or "/mntn" for build.

INSTALL

On target system, here CentOS-5.0 with gcc 4.x:

1. call "make install"

Verify

On the execution-target machine, this is the machine running the actual QEMU-VM, call

```
"ctys-plugins -T all -E"
```

and on the client machine

```
"ctys-plugins -T all"
```

When the last error message complains the absence of QEMUSOCK and QEMUMGMT, than anything seems to be perfect, just missing the final call for network setup by "ctys-setupVDE".

2.3 Setup Networking for QEMU by VDE

2.3.1 Overview

Now, once the "make install" is performed, the base package of QEMU is installed. As mentioned before, here the `qemu` package is not installed, because stacked VMs and heterogeneous CPU environments are going to be tested.

The next step now is to create a runtime environment and install the BIOS binaries and test examples for later validation of QEMU.

The expected runtime structure is as described before:

```
mkdir -p $HOME/qemu/{ctys,bc-bios/keymap,tst}
```

```
cp -a $HOME/.ctys/templates/qemu/tst/* \
  $HOME/qemu/tst
```

Check the `x`-bit of all contained configuration scripts, which are bash-scripts actually to be called as executable.

Now copy and unpack the appropriate test images from the `qemu` distribution into the `tst`-subdirectories.

- `arm-test-0.2.tar.gz`
- `coldfire-test-0.1.tar.bz2`
- `linux-0.2.img.bz2`
- `mipsel-test-0.2.tar.gz`

- mips-test-0.2.tar.gz
- small.ffs.bz2
- sparc-test-0.2.tar.gz

These images are ready-to-use VMs. The only configuration required later is the setting of appropriate IP address, except for the coldfire image, which is based on DHCP. This is only true if you are using DHCP and have set the appropriate MAC addresses of course.

The usage of bridged network with communications via the NIC of the host requires some additional effort. Particularly the creation of the required TAP device with the frequently mentioned "tunctl" utility from the "UserModeM-Linux" was somewhat difficult on CentOS-5. This results particularly from the missing tool "tunctl".

The package "vde" including a (not-found-documentation-for) utility "vde_tunctl" was the rescue-belt.

So proceed as described in the common examples Section 6.1 'TAP/TUN by VDE' on page 35 for your matching environment in order to create the required TAP device and the appropriate "vde_switch".

Now edit the configuration files within the tst subdirectories and adapt the MAC addresses, IP-Addresses, and UUIDs as required. They might be usable as given, if no collisions with existing addresses occur.

At this point anything might be prepared for successful operations.

Now test the installation:

1. raw-call without network for linux-subdirectory.

```
qemu linux-0.2.img
```

2. raw-call with network

```
vdeqemu -vnc :17 -k de -m 512
-hda linux-0.2.img
-net nic,macaddr=<mac>
-net vde,sock=/var/tmp/vde_switch0.<user>
```

Within GuestOS:

```
ifconfig eth0 <your-ip>
ping -c 5 <known-host>
```

Should show what you expect. OK, if so, than you've got it!

3. encapsulation ctys-script-call

```
"/linux.ctys SDL"
```

Might work now.

4. ctys call

2.3.2 VDE

After the "some hours" of trial-and-error the VDE project (see [131, Virtual-Square] and [130, VDE]) was found and installed shortly before giving it up.

After some straight-forward steps now - take due to own mistakes for about 15minutes - anything worked perfectly.

Several combinations of VDE are installed and configured in order to create a TAP device. The various cases are listed due to their generic character within the chapter containing the common examples.

The following cases are shown in the examples section:

- VDE within Xen Dom0
- VDE within Xen DomU
- VDE within VMware
- VDE within Native Unix

2.3.3 QEMU

For Qemu several excellent sites with install descriptions exist in the net. Here are just some shortcuts, which seem to be the most important items.

PXE-Boot

The following applies to a configured PXE environment based on PX-ELinux.

```
vdeqemu -vnc :17 -k de -m 512
        -hda linux-0.2.img
        -net nic,macaddr=<mac>
        -net vde,sock={QEMUSOCK}
        -boot n
        -option-rom ${QEMUBIOS}/pxe-ne2k_pci.bin
        \&
```

ISO-Image-Boot

The following applies to a boot CDROM image for installation.

```
vdeqemu -vnc :17 -k de -m 512
        -hda linux-0.2.img
        -net nic,macaddr=<mac>
        -net vde,sock={QEMUSOCK}
        -boot d
        -cdrom ${QEMUBASE}/iso/install.iso
        \&
```

shared memory

Should be set in "/etc/fstab" as required. This value might be raised, when nested stacks are used, thus the bottom engine requires the sum of the resources of the stack. The entry might look like:

```
none /dev/shm tmpfs defaults,size=512M 0 0
```

For a call like:

```
vdeqemu -m 512 ...
```

The changes could be activated with

```
mount -o remount /dev/shm
```

Create image

To create an ISO image a call like the following could be applied

```
qemu-img create -f qcow myImage.qcow 4G
```

which could be used as

```
qemu -cdrom installMedia.iso \
    -boot d myImage.qcow
```

2.4 Setup Serial Console

The setup of a serial console for QEMU is required for various CONSOLE types. Any CONSOLE providing an ASCII-Interface, except the synchronous un-detachable CLI console, requires serial access to the GuestOS. This is a little complicated to setup for the first time, but once performed successful, it becomes an easy task for frequent use.

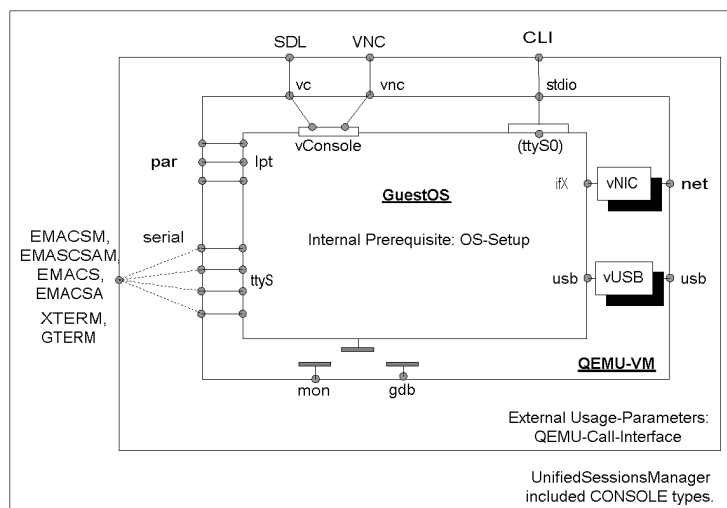


Figure 2.1: QEMU Interconnection-Interfaces

The first thing to consider is the two step setup, which comprises the initial installation with a standard interface either by usage of SDL or by usage of the VNC console. The second step - after finishing the first successfully - is to setup the required serial device within the GuestOS. This requires a native login as root. Detailed information is for example available at "Linux Serial Console HOWTO"[137, VANEMERY], "Serial-HOWTO"[138, GHAWKINS], and "Text-Terminal-HOWTO"[139, DSLAWYER].

The following steps are to be applied.

1. Install GuestOS by usage of SDL/VNC as console.
2. Login into the GuestOS.
3. Adapt `"/boot/grub/menu.lst"`

The header section:

```
serial --unit=0 --speed=9600 --word=8 --parity=no --stop=1
terminal --dumb serial console
splashimage=(hd0,0)/grub/splash.xpm.gz
```

```
#
default=<#yourKernelWithConsole>
#
```

Your target kernel for boot <yourKernelWithConsole>:

```
kernel ... console=tty0 console=ttyS0,9600n8
```

4. Adapt "/etc/inittab", here for CentOS-5.0

```
S0:12345:respawn:/sbin/agetty ttyS0 9600 Linux
```

5. Adapt ctys-qemu-wrapper

This requires adapted items, which depend on the chosen CONSOLE type.

- A CLI, which is a synchronous console, requires:

```
-nographic
```

This switches off the graphic and defaults it's IO to the caller's CLI.

- A "by-Window-Encapsulated-CLI", which is a non-"modal" console, called as a serial console by a UNIX-Domain socket within a X11-Window/Client, requires:

```
...
```

This switches off the graphic and defaults it's IO to the caller's CLI.

- Reboot.

Chapter 3

Installation of Guests

3.1 PXE-Boot

The PXE based installation is possibly not the fastest, but it offers a common neatless solution for unified installation processes. Even though an image could be just copied and modified as required, some custom install procedures might be appreciated, when the install could be performed in batch-mode. One example is the usage of kickstart files for CentOS/RHEL. In case of PXE these files are almost the same for any install base, this spans from physical to virtual machines.

3.2 Install Procedures

3.2.1 Debian-4.0_r3 by PXE - i386

The template "debian-i386" contains particularly the BOOTMODES PXE and INSTALL.

3.2.2 Debian-4.0_r3 for ARM

The installation of debian for ARM9 is quite good described for example by the online articles of Aurelien Jarno[93] and Brian Dushaw[95]. The following description is based on [93, AurelienJarno], which was adapted to the usage of VDE.

For analysis and detailed understanding of the required installprocedures of debian on a non-standard platform for an emulated CPU, this is explained first as a manual procedure based on [93].

- (a) download pre-requisites

Even though the kernel and the initrd could be cross-compiled from the scratch, the prepared variants are used here, these could be downloaded from [94, AurelienJarnoPrecomp], for additional information

refer to [93].

This description is based on the "etch" distribution "debian-4.0_r3". The ISO images are used on a local iso server by http protocol, where the authentication has to be suppressed due to missing "Release.gpg" file. The following files are required (versions may vary) in addition to the ISO images.

- vmlinuz-2.6.16-6-versatile
- initrd.img-2.6.16-6-versatile
- initrd.gz (for install only)

(b) create image for vhdd

```
qemu-img create -f qcow hda.img 10G
```

(c) start installation

The following call is based on the previous setup of VDE by execution of "ctys-setupVDE".

```
vdeq qemu-system-arm -M versatilepb \
  -kernel vmlinuz-2.6.16-6-versatile \
  -initrd initrd.gz \
  -hda hda.img \
  -k de \
  -net nic,macaddr=00:50:56:13:11:42 \
  -net nic,sock=/var/tmp/vde_switch0.tstusr \
  -append "root=/dev/ram \
  -- debian-installer/allow_unauthorized=true"
```

(d) installation steps

The installation itself should be proceeded as described in [93, AurielienJarno] the references. Just some minor issues remain:

- keyboard

During install some keys are not mapped correctly, for the german keyboard within the test environment this is "/", which is actually printed by "Shift-6".
- expected-install errors

Due to missing native kernel, the external kernel and initrd are used by call parameters, for install and runtime. Therefore any error message related to missing kernel on installed system could be ignored safely.
- finish install

The final step has to be "finished" but not "cancelled", and when the VM automatically reboots for first time, it has to be "stopped". Preferably by usage of the QEMU-monitor. When cancelled the installed image may be corrupt and hangs during boot at fsck.

- (e) boot runtime-system
 Once the system is installed, the install-initrd has to be replaced by the runtime specific image and the root device needs to be changed to the runtime image.

The example values has to be adapted, of course.

```
vdeq qemu-system-arm -M versatilepb \
  -kernel vmlinuz-2.6.16-6-versatile \
  -initrd initrd.img2.6.16-6-versatile \
  -hda hda.img \
  -name debian-arm-demo \
  -k de \
  -net nic,macaddr=00:50:56:13:11:42 \
  -net nic,sock=/var/tmp/vde_switch0.tstusr \
  -append "root=/dev/sda1"
```

3.2.3 OpenBSD-4.3+SerialConsole - by PXE

The installation is straight forward(refer to template for tst127, BOOT-MODE=PXE). When for later access a serial console is required following additional steps has to be proceeded. A serial console is a prerequisite for EMACS, XTERM, and GTERM.

- (a) Boot and login into Guest OS if the root-filesystem cannot be mounted offline.
- (b) Create a file `"/etc/boot.conf"` with content such as:

```
set tty com0
stty com0 115200
set timeout 15
boot
```

- (c) Edit the file `"/etc/tty"` and change the lines:

```
tty00  "/usr/libexec/getty std.115200" vt220 on secure
```

3.3 Installed Systems

OS	name	Inst-VM
CentOS-5.0	tst000	0.9.1
Debian-4.0r3-arm	tst102	0.9.1
Debian-4.0r3-i386	tst108	0.9.1
Fedora 8	tst105	0.9.1, 0.9.0
OpenBSD-4.0	tst124	0.9.1
OpenBSD-4.3	tst127	0.9.1
SuSe-10.3	tst113	0.9.1
uClinux-arm9	(QEMU)	0.9.1
uClinux-coldfire	(QEMU)	0.9.1

Table 3.1: Overview of Installed-QEMU-VMs

In addition various test packages with miscellaneous CPU emulations of QEMU are available. Example templates for integration scripts are provided for ARM, Coldfire, MIPS, and PPC.

Chapter 4

Call Examples

4.1 CREATE a session

The first tests and examples of the QEMU plugin are based on the "arm-tst" VM contained in the examples of QEMU. This is a ready to use VM, but the TCP/IP address is hardcoded to "10.0.0.2" thus might be required to be configured. The coldfire test VM contained in the QEMU examples supports DHCP, thus is ready to use within the network.

Anyhow, for the first tests the actual usage of the network is not yet required. All following examples, if not stated else, rely on the provided configuration file "arm-test.ctys" and the QEMU VM "arm-test". Theses have to be installed as described within the examples chapter Section 2 'Installation of QEMU' on page 13 .

The first call now creates a session and starts the VM with VNC as a console which will be attached automatically.

```
ctys \  
-t qemu \  
-a  
create=p:$HOME/qemu/tst/arm-test/arm-test.ctys,\  
reuse,console:vnc  
lab00
```

When the "vde_switch" is not configured yet the following error message occurs:

```
Missing management socket for "vde_switch"  
QEMUMGMT=/var/tmp/vde_mgmt0.acue  
Call: "ctys-setupVDE" on "lab00.soho"
```

The solution is simply to proceed as requested and just create the UNIX Domain sockets by the following call:

```
ctys-steupVDE -u $USER create
```

This call requires root permissions.

That's it.

The support of the types of CONSOLES depends on the actually implemented call within the "arm-tst.ctys" script, which is a shell script with a defined interface. The currently supported CONSOLE types by arm-test are: "CLI, SDL, VNC". The CLI and SDL types are supported as DISPLAYFORWARDING in synchronous mode only for this version.

The following call creates a SDL CONSOLE.

```
ctys \  
-t qemu \  
-a  
create=p:$HOME/qemu/tst/arm-test/arm-test.ctys,\  
reuse,console:sdl  
lab00
```

As might be expected, the following call creates a CLI CONSOLE.

```
ctys \  
-t qemu \  
-a  
create=p:$HOME/qemu/tst/arm-test/arm-test.ctys,\  
reuse,console:cli  
lab00
```

The monitor as configured within "arm-test.ctys" could be attached by usage of "netcat"

```
nc -U ${MYQEMUMONSOCK}
```

which could be generated by function "netGetUNIXDomainSocket" and is derived from "QEMUMONSOCK" as raw-pattern, for additional information refer to the "arm-text.ctys" inline comments.

The QEMU monitor now could be entered by typing "Ctrl-a c RET", the console is recovered by typing the same sequence again. For additional information refer to the QEMU User-Manual.

A second terminal emulation to be used is the "unixterm" command of VDE.

Alternatively EMACS could be used as a terminal emulation for CONSOLE access, either in "shell-mode" or in "ansi-term" mode. This works the same way as an ordinary xterm session, where within the "display-window" a cli is started connecting to a local UNIX domain socket. The socket has to be configured as a serial device within the GuestOS. For EMACS two variants additional variants exist for both modes, where the frame is divided into two windows, which connect the <execution-target> and the <machine-address> representing the GuestOS.

In the following example in the upper window a login-prompt of the GuestOS is displayed, whereas in the bottom window the "top" command is shown for the hosting machine.

```

CentOS release 5 (Final)
Kernel 2.6.18-0.el5 on an i686

tst000.soho login: set_rtc_mmss: can't update from 0 to 59
[]

-----
su*** *tst000* All (7,0) (Term: char run Abbrev)-----
top - 17:50:45 up 2 days, 15:38, 3 users, load averages: 3.19, 3.54, 4.49
Tasks: 423 total, 9 running, 414 sleeping, 0 stopped, 0 zombie
Cpu(s): 41.1%us, 24.0%sy, 0.0%ni, 34.6%id, 0.0%wa, 0.0%hi, 0.3%si, 0.0%st
Mem: 3995940k total, 396040k used, 35432k free, 38020k buffers
Swap: 20466768k total, 125532k used, 20341236k free, 2616352k cached

-----
su*** *app2.soho* All (6,0) (Term: char run Abbrev)-----

```

PID	USER	PR	NI	VTID	RES	SHR	S	%CPU	MEM	TIME+	COMMAND
24793	acue	5	-10	699m	411m	398m	S	0	10.5	1:23.67	vmware-vmx
25371	acue	15	0	699m	411m	398m	S	0	10.5	0:00.24	vmware-vmx
25411	acue	5	-10	699m	411m	398m	S	0	10.5	0:02.02	vmware-vmx
25417	acue	15	0	699m	411m	398m	S	2	10.5	3:25.60	vmware-vmx
25567	acue	15	0	699m	411m	398m	S	0	10.5	0:00.22	vmware-vmx
25625	acue	15	0	699m	411m	398m	S	0	10.5	0:00.26	vmware-vmx
26335	acue	15	0	699m	411m	398m	S	0	10.5	0:00.23	vmware-vmx
27212	acue	15	0	699m	411m	398m	S	0	10.5	0:00.21	vmware-vmx
27223	acue	15	0	699m	411m	398m	S	0	10.5	0:00.26	vmware-vmx
27224	acue	15	0	699m	411m	398m	S	0	10.5	0:00.23	vmware-vmx
27307	acue	15	0	699m	411m	398m	S	0	10.5	0:00.25	vmware-vmx
27370	acue	15	0	699m	411m	398m	S	0	10.5	0:00.22	vmware-vmx

Figure 4.1: The "CONSOLE:EMACSAM" for a QEMU Session

The console with pure CLI access could be combined with an VNC console allowing additional graphical access. This is particularly forseen, and will be offered soon, ad a debugging facility for GDB access to QEMU and to applications within the GuestOS.

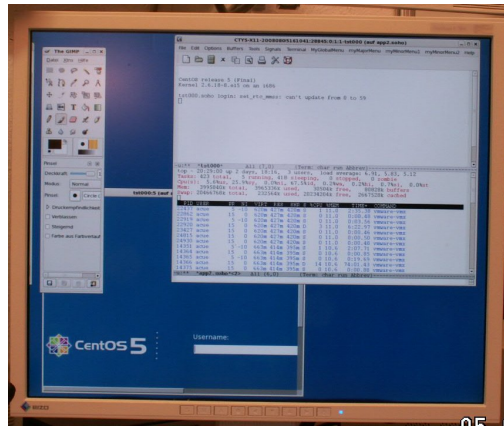


Figure 4.2: The "CONSOLE:EMACSAM" and a "CONSOLE:VNC" for a QEMU Session

4.2 CANCEL a session

The following call CANCELs the arm-test session.

```
ctys \  
-t qemu \  
-a cancel=p:$HOME/qemu/tst/arm-test/arm-test.ctys,\  
force,poweroff:0\  
lab00
```

4.3 LIST sessions

The following call LISTs all sessions:

TCP-container	TCP-guest	label	sesstype	c	user	group
lab00.soho	-	arm-test	VNC	C	acue	ldapusers
lab00.soho	-	LAB00	VNC	C	root	root
lab00.soho	-	LAB00	VNC	S	root	root
lab00.soho	tst109	arm-test	QEMU-arm	S	acue	ldapusers
lab00.soho	-	Domain-0	XEN	S	-	-
lab00.soho	lab00.soho.	lab00	PM	S	-	-

The following call LISTs all sessions by usage of a specific LIST-MACRO for QEMU:

```
ctys macro:listconnpid lab00
```

Resulting in:

label	stype	c	DIS	cport	sport	pid	PM	TCP
LAB00	VNC	C	1			18933	lab00.soho	
LAB00	VNC	S	1	5901		5642	lab00.soho	
arm-test	QEMU-arm	S	17		25704	25832	lab00.soho	
Domain-0	XEN	S					lab00.soho	
lab00	PM	S				1	lab00.soho	192.168.1.71

4.4 ENUMERATE sessions

The following call ENUMERATEs all stored session configurations within the subdirectory of the HOME.

```
ctys -t qemu -a enumerate=b:qemu/tst lab00
```

The following call displays a listing formatted as a table:

```
ctys -t qemu macro:listconn lab00
```

4.5 SHOW

The following call SHOWs dynamic data.

```
ctys -t qemu -a show lab00
```

4.6 INFO

The following call display static data. Particularly the available capabilities for QEMU are displayed, which contains a list of all available CPUs and the related system boards.

```
ctys -t qemu -a info lab00
```

This leads to:

```
#####
Node:lab00.soho - ctys(01_04_001A03)
System      :Linux
OS          :Linux
RELEASE     :2.6.18-8.1.15.el5.centos.plusxen
MACHINE     :x86_64
KERNEL#CPU  :SMP-KERNEL
CPU-INFO
  processor:0
    vendor_id      :GenuineIntel
    cpu family     :6
    model          :22
    model name     :Intel(R) Celeron(R) CPU 420 @ 1.60GHz
    stepping      :1
    cpu MHz        :1599.853
    cache size     :512 KB

  Flags assumed equal for all processors on same machine:
  flags
    vmx(VT-x - Pacifica) = 0
    svm(AMD-V - Vanderpool) = 0
    PAE = 1

MEM-INFO
  MemTotal      : 523 M
  SwapTotal     : 2031 M

SOFTWARE
Mandatory:
  bash          :GNU bash, version 3.1.17(1)-release
                (x86_64-redhat-linux-gnu)
  gawk          :GNU Awk 3.1.5
  sed          :GNU sed version 4.1.5
  SSH          :OpenSSH_4.3p2, OpenSSL 0.9.8b 04 May 2006
  top          :top: procps version 3.2.7

Optional:
  wmctrl       :wmctrl is on this machine not available
  lm_sensors   :sensors version 2.10.0 with libsensors version
                2.10.0
  hddtemp      :hddtemp version 0.3-beta13

PLUGINS      :QEMU CLI X11 VNC
QEMU:        Plugin Version:01.01.001a00pre
              Operational State:ENABLED
              QEMU version:
              ->QEMU-0.9.1
                Magic-Number:QEMU_091
                Verified Prerequisites:
                ->CLI-ValidatedBy(hookInfoCheckPKG)
                ->X11-ValidatedBy(hookInfoCheckPKG)
                ->VNC-ValidatedBy(hookInfoCheckPKG)
                -><LocalClientCLI>
                -><LocalClientX11>
                -><LocalClientVNC>
                -><LocalXserverDISPLAY>
                -><delayedValidationOnFinalTarget>
```

```

-><QEMU-0.9.1>
->_/_usr/local/bin/vde_switch_info-USER=
    acue-ACCESS-PERMISSION-GRANTED
->_/_usr/local/bin/unixterm_info-USER=
    acue-ACCESS-PERMISSION-GRANTED
->_/_usr/local/bin/vdeq_info-USER=
    acue-ACCESS-PERMISSION-GRANTED
->_/_usr/local/bin/vdeqemu_info-USER=
    acue-ACCESS-PERMISSION-GRANTED
-><QEMUSOCK=/var/tmp/vde_switch0.acue_info-USER=
    acue-ACCESS-GRANTED>
-><QEMUMGMT=/var/tmp/vde_mgmt0.acue_info-USER=
    acue-ACCESS-GRANTED>
-><CPU-Emulation:qemu-alpha>
-><CPU-Emulation:qemu-arm>
-><CPU-Emulation:qemu-armeb>
-><CPU-Emulation:qemu-cris>
-><CPU-Emulation:qemu-i386>
-><CPU-Emulation:qemu-img>
-><CPU-Emulation:qemu-m68k>
-><CPU-Emulation:qemu-mips>
-><CPU-Emulation:qemu-mipsel>
-><CPU-Emulation:qemu-ppc>
-><CPU-Emulation:qemu-ppc64>
-><CPU-Emulation:qemu-ppc64abi32>
-><CPU-Emulation:qemu-sh4>
-><CPU-Emulation:qemu-sh4eb>
-><CPU-Emulation:qemu-sparc>
-><CPU-Emulation:qemu-sparc32plus>
-><CPU-Emulation:qemu-sparc64>
-><CPU-Emulation:qemu-system-arm>
-><CPU-Emulation:qemu-system-cris>
-><CPU-Emulation:qemu-system-m68k>
-><CPU-Emulation:qemu-system-mips>
-><CPU-Emulation:qemu-system-mips64>
-><CPU-Emulation:qemu-system-mips64el>
-><CPU-Emulation:qemu-system-mipsel>
-><CPU-Emulation:qemu-system-ppc>
-><CPU-Emulation:qemu-system-ppc64>
-><CPU-Emulation:qemu-system-ppcemb>
-><CPU-Emulation:qemu-system-sh4>
-><CPU-Emulation:qemu-system-sh4eb>
-><CPU-Emulation:qemu-system-sparc>
-><CPU-Emulation:qemu-system-x86_64>
-><CPU-Emulation:qemu-x86_64>

```

```

CLI:      Plugin Version:01.01.001a02
          Operational State:DISABLED

X11:     Plugin Version01.01.001a02
          Operational State:DISABLED

VNC:     Plugin Version:01.02.001b01
          Operational State:DISABLED

```


Chapter 5

Example Appliances

5.1 Qemu Test and Demo Examples

5.1.1 ARM

After installation of QEMU and VDE as described in Section 2 [‘Installation of QEMU’](#) on page 13 for validation of the operational state of QEMU the utility "ctys-plugins" should be called. The following call verifies the different plugins operational states for server functionality.

```
ctys-plugins -T all -e
```

The client functionaity could be verified with the call:

```
ctys-plugins -T all
```

Now, with a properly installed test environment from QEMU and the additional ctys call-scripts setup as described before, the following call should start the "arm-test" QEMU VM with and CONSOLE of type SDL.

```
ctys \  
-t qemu \  
-a create=f:qemu/tst/arm-test/arm-test.ctys,\  
  console:sdl \  
lab00
```

In case of ambiguous filenames in the cacheDB e.g. due to multiple access paths on multiple nodes by NFS the following approaches could be applied

- use "p:<pathname>"
When the full absolute path by "p:<pathname>" is provided, no local ambiguity may occur within the execution context.

This is recommended for the first steps, because it does not require any additional action.

- extend number of params from <machine-address>
Additional entries may lead to unambiguity. This depends on the contents of the distributed caches and requires some deeper knowledge of the system. Even though this is not that much, it should be shifted when doing the first steps.
- deactivate cacheDB
Another quick solution is the disabling of any caching, therefore the options "-c off" and "-C off" could be set. This leads to a filesystem scan, which of course results in some performance degradation, which could be serious in case of deep filestructures with a "late match". The scan is performed by usage of the system utility "find".

The supported CONSOLE types for the from-the-box "arm-test" VM are CLI, SDL, and VNC. Additional information is available as inline comment within the "arm-test.ctys" configuration from the

```
"$HOME/ctys/templates"
```

directory.

After this call an SDL terminal window should be opened.

In case of networking problems the most common error is the forgotten call of "ctys-setupVDE -u <USER> create".

5.1.2 Coldfire

```
ctys \  
-t qemu \  
-a create=f:/qemu/tst/coldfire-test-0.1/coldfire.ctys,\  
  console:cli\  
lab00
```

Chapter 6

System Resources

6.1 TAP/TUN by VDE

A TAP device in combination with virtual switches will be applied in accordance to the introduction within Section ?? ‘??’ on page ?? . The actual implementation is based on the package "*Virtual Distributed Ethernet*" - *VDE(vde2)*[130, sourceforgeVde].

The only and main restriction was the originally required presence of the package "userspace-utils" from the UML project, containing the "tunctl" tool for tap creation and assignment of it's owner. This is, at least not out-of-the-box available for the majority of distributions, but within "*Virtual Distributed Ethernet*", by the time of writing this dokument it is not yet documented.

The tool called "vde_tunctl" from "*Virtual Distributed Ethernet*" could be used as a perfect replacement. The additional benefit is the included "vde_switch", which is a virtual switch, able to operate in hub-mode. This supports user-space access and on demand creation of new interfaces with non-privileged users permission only. Just one initial TAP device is required to be created with root permissions.

Thus the following basic structure is assumed to be pre-configured for each user and interconnected to the main virtual-bridge:

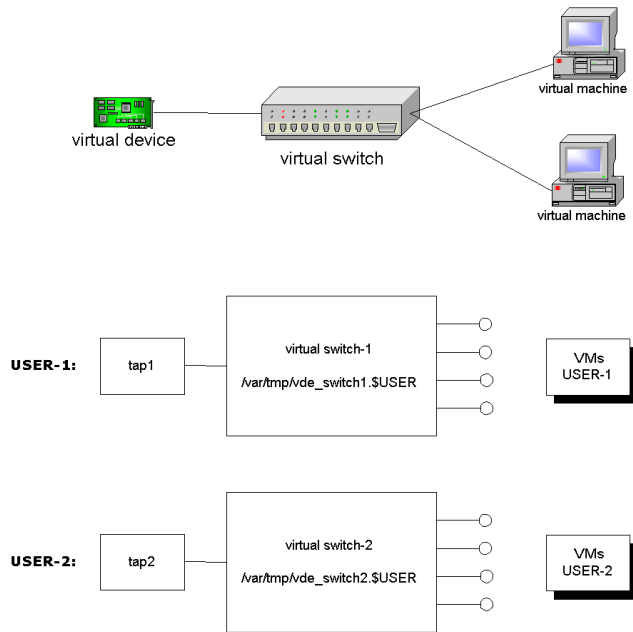


Figure 6.1: Virtual switch by vde_switch

Thus each user allowed to access external network has it's individual virtual switch with user-space access.

6.1.1 VDE within Xen Dom0

This lists required steps to setup a QEMU instance within a Xen-Dom0 VM using the default xenbr0 bridge for external interconnection. The most of the required steps are covered within `ctys-setupVDE`. Due to required root-permissions for some of called system tools appropriate `sudo/ksu` configuration should be in place. For the display of a listing of called tools refer to "`ctys-plugins`".

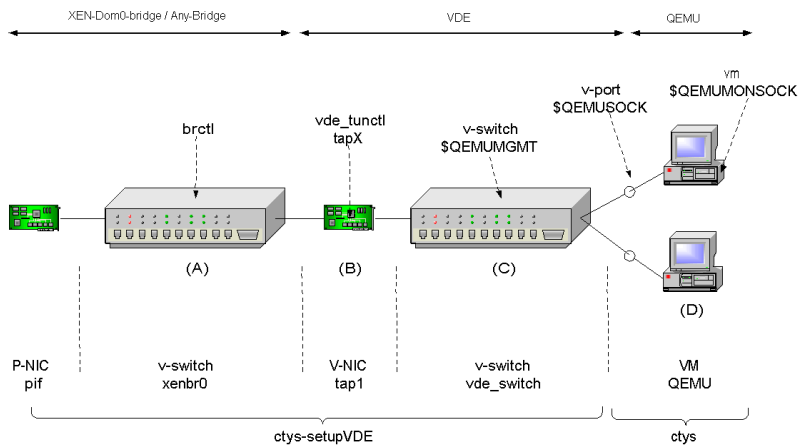


Figure 6.2: QEMU NIC interconnection

- (a) For "*CentOS-5.0*" [74, centOS] the "*vde2-2.2.0-pre1 version*" [130, sourceforgeVde] seems to work perfectly.

Just **download** [130, sourceforgeVde] it and follow the instructions. For the original distribution a "configure+make+make install" is required.

- (b) Prepare networking for Qemu by calling

```
ctys-setupVDE -u <switch-owner> create
```

This call creates a new switch to be used e.g. by QEMU, where <switch-owner> is the user which owns the communications sockets to the switch, default is "\$USER".

Even though it is possible to use `sudo` and/or `ksu` for `ctys-setupVDE`, this is not recommended, because e.g. `chmod` and `chown` are required in order to change the owner of originally root-owned switch sockets. Thus the creation of the bridge and user-switches should be performed by the administrator as root, or as a admin-only account.

This tool manages the complete network infrastructure required for QEMU and creates required bridges and switches as required. Thus the user does not need to care for the pre-configuration of the network resources.

The following network configurations are supported for "bridging" external NICs.

- single NIC, without a preconfigured bridge

- single NIC, with a preconfigured bridge, e.g. by Xen
- multiple interfaces as a bonding device to be bridged
- multiple interfaces as a bonding device with a preconfigured bridge, e.g. by Xen

The `ctys-setupVDE` utility can be applied within Xen-Dom0, Xen-DomU, VMware, QEMU, and native; additional options are available.

(c) Call QEMU with a VDE wrapper

The final call from within `ctys` should be wrapped in accordance to internal call interface, the "raw" call shown here is just for an initial test by provided QEMU-Example VMs. `ctys-wrapper` for QEMU-Examples are provided within the installed `"$HOME/ctys/templates/qemu-examples-wrapper"` subdirectory.

The variable `QEMUSOCK` could be configured within `qemu.conf.$MYOS`.

```
QEMUSOCK=/var/tmp/vde_switch0.$USER
```

```
vdeqemu -vnc :17 \
  -k de \
  -net nic,mac=<MAC-addr> \
  -net,sock=${QEMUSOCK} \
  linux-0.2.img \&
```

(d) Attach a vncviewer

```
vncviewer :17&
```

(e) Connect GuestOS to external network

```
ifconfig eth0 <IP-address>
```

The default address is something like "10.0.2.15", when your network fits to this, nothing is required to be done.

(f) Call `ssh`, and or `ping`, or whatever you want.

That's it.

6.1.2 VDE within Xen DomU

See Section 6.1.1 '[VDE within Xen Dom0](#)' on page 36 .

6.1.3 VDE within VMware

See Section 6.1.1 '[VDE within Xen Dom0](#)' on page 36 .

6.1.4 VDE within Native Unix

These are the required steps to setup a QEMU instance on a native Linux system creating a new bridge with `brctl` for external interconnection. The most of the required steps are covered within `ctys-setupVDE`. Due to required root-permissions for some of called system tools appropriate `sudo/ksu` configuration should be in place. For the display of a listing of called tools refer to Section ?? '??' on page ?? .

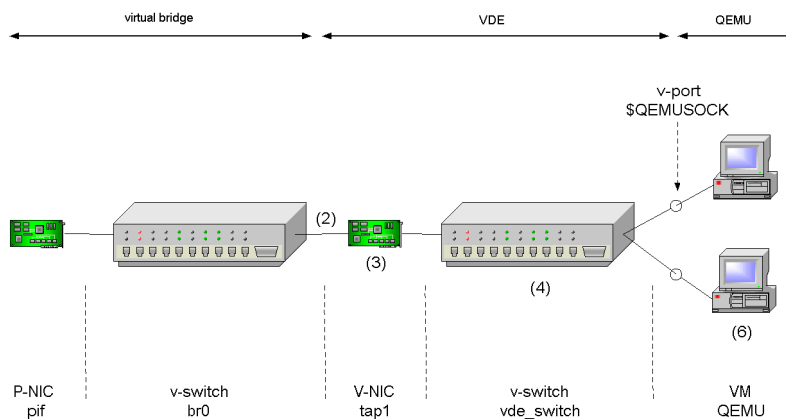


Figure 6.3: QEMU NIC interconnection for Native UNIX

Pre-required admin tasks with root permissions, not called within `ctys`.

- (a) For "*CentOS-5.0*"[\[74, centOS\]](#) the "*vde2-2.2.0-pre1 version*"[\[130, sourceforgeVde\]](#) seems to work perfectly.

Just download it and follow the instructions. For the original distribution a "configure+make+make install" is required.

- (b) Create: root permission required
"tapX" with

```
"vde_tunctl -u <user-without-root-permission>"
```

e.g.

```
vde_tunctl -u acue
```

Returns a line like:

```
"Set 'tap3' persistent and owned by uid 4711"
```

- (c) Call: root permission required
Create a new bridge:

```
brctl addbr br0
```

- (d) Call: root permission required
Disable STP for now:

```
brctl stp br0 off
```

- (e) Call: root permission required
Add interface - bonding interface here:

```
brctl addif br0 bond0
```

- (f) Call: root permission required
Deactivate bonding interface:

```
ifconfig bond0 down
```

- (g) Call: root permission required
Set to promiscuous mode:

```
ifconfig bond0 0.0.0.0 up
```

- (h) Call: root permission required
Set IP address for host:

```
ifconfig br0 <ip-of-bond0> up
```

Call ping, but may take some trials and some seconds - 4-5 fails within about 20-40seconds, than works.

- (i) Call: root permission required
Use the returned 'tap3' for following call:

```
/etc/xen/qemu-ifup tap3 br0
```

This is due to the usage of QEMU within Xen, for the targeted stack test. Any other configuration will be similar to this.

The given script could be at various locations, it contains basically the following calls only:

```
ifconfig $1 0.0.0.0 up
brctl addif $2 $1
```

Which brings up the newly created interface 'tap3' and adds an interface to the virtual Xen bridge connecting it to the world outside.

The results could be verified with:

```
ifconfig tap3
  should list an interface 'tap3'
brctl show
  should contain an interface 'tap3'
```


- (j) Call: root permission required

Now this interface will be connected to another virtual switch, the `vde_switch` in order to provide an internal multiplexer for multiple QEMU instances to be connected to the external interfaces via the Xen-bridge.

```
QEMUSOCK=/var/tmp/vde_switch0.$USER
QEMUMGMT=/var/tmp/vde_mgmt0.$USER
```

```
vde_switch -d \
  -tap tap3 \
  -s ${QEMUSOCK} \
  -M ${QEMUMGMT}
```

```
chown -R <userX.groupX> ${QEMUSOCK}
chown -R <userX.groupX> ${QEMUMGMT}
```

- (k) Call: optional

```
QEMUMGMT=/var/tmp/vde_mgmt0.$USER
```

```
unixterm ${QEMUMGMT}
```

User tasks, not requiring root permissions, called within `ctys`.

- (l) Call:

The variable `QEMUSOCK` could be configured within `qemu.conf.$MYOS`.

```
QEMUSOCK=/var/tmp/vde_switch0.$USER
```

```
vdeqemu -vnc :17 \
  -k de \
  -net nic,mac=<MAC-addr> \
  -net,sock=${QEMUSOCK} \
  linux-0.2.img \&
```

- (m) Call:

```
vncviewer :17\&
```

- (n) Call within GuestOS:

```
ifconfig eth0 <IP-address>
```

The default address is something like "10.0.2.15", when your network fits to this, nothing is required to be done.

- (o) Call
- `ssh`
- , and or
- `ping`
- , or whatever you want.

That's it.

6.1.5 VDE Remote Configuration

The tool `ctys-setupVDE` supports the remote configuration and management of virtual VDE based switches. Due to the supported call with a

list, multiple targets could be configured by one call. The required access permissions need to be pre-configured of course.

The following call creates on the hosts: "host1, host2, host3" a VDE-Switch and if not yet present a "Main Virtual-Bridge". The user is as default the caller's ID, here "root".

```
ctys-setupVDE create root@host1 root@host2 root@host3
```

The same could be performed by any user, but requires than for creation release access to system resources for the following tools(paths may vary) either by ".k5users" or "sudoers":

- /usr/local/sbin/vde_tunctl
- /sbin/ifconfig
- /usr/sbin/brctl

Than the same call e.g. looks like:

```
ctys-setupVDE \
-u charly \
create \
host1'(-Z ksu)' \
host2'(-Z sudo)' \
host3'(-Z ksu)'
```

The following call lists all present switches on all hosts, including the dynamically allocated sockets.

```
ctys-setupVDE listall host1 host2 host3
```

The following call lists all present switches on all hosts, excluding the dynamically allocated sockets.

```
ctys-setupVDE list host1 host2 host3
```

The following call lists the ports of the switches owned by "root".

```
ctys-setupVDE ports root@host1 root@host2 root@host3
```

The following call lists the ports of the switch owned by "charly".

```
ctys-setupVDE \  
  ports \  
    -u charly \  
    root@host1 \  
    root@host2 \  
    root@host3
```

The following call displays some basic information.

```
ctys-setupVDE info root@host3
```

The following call removes the previously created switches and eventually created "Main Virtual-Bridges", if they are no longer in use.

```
ctys-setupVDE \  
  create \  
    root@host1 \  
    root@host2 \  
    root@host3
```

Due to the remote call support of the tools `ctys-genmconf` and `ctys-plugins` the configured information could be verified for the various plugins easily. The required calls are:

```
ctys-plugins -T all -E root@host1 root@host2 root@host3
```

and for actual modification of the PM, either actually a physical machine

```
ctys-genmconf -P PM root@host1 root@host2 root@host3
```

or a virtual stack-entity, but appearing as physical for it's containees

```
ctys-genmconf -P VM root@host1 root@host2 root@host3
```

6.1.6 VDE-Setup with Micro-VMSTACK

The setup procedure of the required network environment based on "ctys-setupVDE" could be automated by usage of a "micro-VMSTACK" by defining either a macro or a synchronous subgroup and/or STACK mode. This is based on the remote-execution-call feature of the standard plugin CLI, which supports the remote execution of library functions and external executables. The reason for the usage of CLI as an execution framework instead of the remote call capability of "ctys-setupVDE" (which is based on CLI), is simply the neatless integration of STACK aware synchronicity, which is supported by the framework for plugin calls within ctys only.

A typical application of this feature within a full sized VMSTACK is shown in the VMStacks.

```
completeQEMUEXample = VMSTACK'\
#####
#0.) Synchronous pre-creation of required TAP device
#   for a QEMU instance.
#   A bridge is created when missing, additionally
#   a switch for non-privileged USER-SPACE usage
#   by user.
#####
root@tst104( \
  -t cli \
  -Z ksu \
  -a create=1:tstVDE,cmd:ctys-setupVDE%-f%-u%acue%create \
) \

#####
#1.) Creation of a VM based on QEMU.
#####
tst104( \
  -t qemu \
  -Z ksu \
  -a create=1:tst127,reuse,console:none,user:woll \
  -c local \
) \
}'
```

6.2 Mount an ISO-Image

Loopback on Linux

Mount an FDD image (from "man mount"):

```
mount /tmp/fdimage /mnt \  
-t msdos \  
-o loop=/dev/loop3,blocksize=1024
```

Mount an CD/DVD image

```
mount <ISO-image-pathname> <mount-point> \  
-t iso9660 \  
-o loop[=/dev/loop<free #dev>[,blocksize=1024]]
```

When "-o loop" provided only, the system will use appropriate defaults. The actual values could be inspected by "mount", the "losetup" call inspects loop devices.

Loopback on OpenBSD

Bibliography

Books

UNIX

- [1] Juergen Gulbins: *UNIX Version 7, bis System V.3*, Springer-Verlag Berlin Heidelberg; 1988; ISBN: 3-540-19248-4
- [2] Maurice J. Bach: *The Design Og The UNIX Operating System*, Prentice Hall, Inc.; 1986; ISBN: 0-13-201757-1
- [3] Sebastian Hetze, Dirk Hohndel, Martin Mueller, Olaf Kirch: *LINUX Anwender Handbuch*, LunetIX Softair; 1994; ISBN: 3-929764-03-2
- [4] Rob Flickenger: *LINUX SERVER HACKS*, O'Reilly&Associates, Inc.; 2003; ISBN: 0-596-00461-3
- [5] Olaf Kirch: *LINUX Network Administrator's Guide*, O'Reilly&Associates, Inc.; 1995; ISBN: 0-56592-087-2
- [6] Daniel P. Bovet, Marco Cesati: *Understanding the LINUX KERNEL*, O'Reilly&Associates, Inc.; 2003; ISBN: 0-596-00002-2
- [7] Daniel P. Bovet, Marco Cesati: *Understanding the LINUX KERNEL(2nd. Ed.)*, O'Reilly&Associates, Inc.; 2003; ISBN: 0-596-00213-0
- [8] Wolfgang Mauerer: *Linux Kernelarchitektur - Konzepte, Strukturen und Algorithmen von Kernel 2.6*, Carl Hanser Verlag Muenchen-Wien; 2004; ISBN: 3-446-22566-8
- [9] Alessandro Rubini: *LINUX Device Drivers*, O'Reilly&Associates, Inc.; 1998; ISBN: 1-565592-292-1
- [10] Alessandro Rubini, Jonathan Corbet: *LINUX Device Drivers(2nd. Ed.)*, O'Reilly&Associates, Inc.; 2001; ISBN: 0-596-00008-1
- [11] Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hartman: *LINUX Device Drivers(3nd. Ed.)*, O'Reilly&Associates, Inc.; 2005; ISBN: 0-596-00590-3
- [12] Juergen Quade, Eve-Katharina Kunst: *Linux-Treiber entwickeln*, dpunkt.verlag GmbH; 2004; ISBN: 3-89864-238-0
- [13] Wehrle, Paehlke, Ritter, Mueller, Bechler: *Linux Netzwerkar-chitektur - Design und Implementierung von Netzwerkprotokollen im Linux-Kern*, Addison Wesley, Inc.; 2002; ISBN: 3-8273-1509-3

- [14] Brandon Palmer, Jose Nazario: *Secure Architectures with OpenBSD*, Addison Wesley, Inc.; 2004; ISBN: 0-321-19366-0
- [15] Michael W. Lucas: *Absolute OpenBSD - UNIX for the Practical Paranoid*, No Starch Press, Inc.; 2003; ISBN: 1-886411-99-9
- [16] Chris Tyler: *FEDORA LINUX*, O'Reilly&Associates, Inc.; 2006; ISBN: 1-596-52682-2
- [17] Benjamin Mako Hill, Jono Bacon, Corey Burger, Jonathan Jesse, Ivan Krstic: *The Official ubuntu Book*, Pearson Education, Inc.; 2007; ISBN: 0-13-243594-2
- [18] Jonathan Oxaer, Kyle Rankin, Bill Childers: *UBUNTU Hacks - Tips&Tools for Exploring, Using, and Tuning Linux*, O'Reilly&Associates, Inc.; 2006; ISBN: 1-596-52720-9
- [19] Tim SchÄ¼rmann: *(K)Ubuntu - Installieren - Einrichten - Anwenden*, Open Source Press; 2006; ISBN: 3-937514-30-9
- [20] Michael Urban, Brian Thiemann: *FreeBSD 6 Unleashed*, Sams Publishing, Inc.; 2006; ISBN: 0-672-32875-5
- [21] Marshall Kirk McKusick, George V. Neville-Neil: *The Design and Implementation of the FreeBSD 6 Operating System*, Addison Wesley, Inc.; 2005; ISBN: 0-201-70245-2

Security

- [22] Guenter Schaefer: *Netzsicherheit - Algorithmische Grundlagen und Protokolle*, dpunkt.verlag GmbH; 2003; ISBN: 3-89864-212-7
- [23] Alexande Geschonneck: *Computer Forensik - Systemeintrübe erkennen, ermitteln, aufklären*, dpunkt.verlag GmbH; 2004; ISBN: 3-89864-253-4
- [24] Jonathan Hassel: *RADIUS - Securing Public Access to Privaze Resources*, O'Reilly&Associates, Inc.; 2002; ISBN: 0-596-00322-6
- [25] Jason Garman: *Kerberos - The Definitive Guide*, O'Reilly&Associates, Inc.; 2003; ISBN: 1-596-00403-6
- [26] Daniel J. Barret, Richard E. Silverman & Robert G. Byrnes: *SSH The Secure Shell - The Definitive Guide*, O'Reilly&Associates, Inc.; 2005; ISBN: 1-596-00895-3
- [27] John Viega, Matt Messier & Pravir Chandra: *Network Security with OpenSSL*, O'Reilly&Associates, Inc.; 2002; ISBN: 0-596-00270-X
- [28] Jonathan Hassel: *RADIUS*, O'Reilly&Associates, Inc.; 2002; ISBN: 0-596-00322-6
- [29] Gerald Carter: *LDAP - System Administration*, O'Reilly&Associates, Inc.; 2003; ISBN: 1-596592-491-6
- [30] Matthias Reinwarth, Klaus Schmidt: *Verzeichnisdienste - Telekommunikation Aktuell*, VDE Verlag GmbH; 1999; ISBN: 3-8007-2373-5
- [31] Dieter Klüenter, Jochen Laser: *LDAP verstehen, OpenLDAP einsetzen*, dpunkt.verlag GmbH; 2003; ISBN: 3-89864-217-8

- [32] Daniel J. Barret, Richard E. Silverman & Robert G. Byrnes: *LINUX Security Cookbook*, O'Reilly&Associates, Inc.; 2003; ISBN: 1-565-00391-9
- [33] Charlie Scott, Paul Wolfe & Mike Erwin: *Virtual Private Networks*, O'Reilly&Associates, Inc.; 1999; ISBN: 1-565592-529-7
- [34] Bill McCarty: *SELINUX - NSA's Open Source Security Enhanced Linux*, O'Reilly&Associates, Inc.; 2004; ISBN: 1-565-007161-7
- [35] Rober L. Ziegler: *Linux Firewalls*, New Riders Publishing, Inc.; 2000; ISBN: 0-7357-0900-9
- [36] D. Brent Chapman and Elisabeth D. Zwicky: *Einrichten von Internet Firewalls*, O'Reilly&Associates, Inc.; 1996; ISBN: 3-930673-31-2
- [37] Wolfgang Barth: *Das Firewall-Buch*, Nicolaus Millin Verlag GmbH; 2004; ISBN: 3-89990-128-2
- [38] Joseph Kong: *Designing BSD Rootkits - An Introduction to Kernel Hacking*, No Starch Press, Inc.; 2007; ISBN: 978-1-593271-142-8
- [39] Cyrus Peikari, Anton Chuvakin (Peter Klicman, Andreas Bildstein, Gerald Richter): *Kenne deinen Feind - Fortgeschrittene Sicherheitstechniken*, O'Reilly&Associates, Inc.; 2004; ISBN: 3-89721-376-1
- [40] Ryan Russel et al.: *Die mitp-Hacker-Bibel*, mitp-Verlag/Bonn; 2002; ISBN: 3-8266-0826-3
- [41] Wallace Wang: *Steal This Computer Book 4.0 - What They Won't Tell You About The Internet*, No Starch Press, Inc.; 2007; ISBN: 1-59327-105-0
- [42] Johnny Long: *Google Hacking - For Penetration Testers*, Syngress Publishing, Inc.; 2005; ISBN: 1-931836-36-1
- [43] Thomas Bechtold, Peter Heinlein: *Snort, Acid & Co. - Einbruchserkennung mit Linux*, Open Source Press; 2004; ISBN: 3-937514-01-3
- [44] Syngress Autorenteam: *Snort 2.0 Intrusion Detection*, mitp-Verlag/Bonn; 2003; ISBN: 3-6266-1304-X

Networks

- [45] Charles E. Spurgeon: *Ethernet - The Definitive Guide*, O'Reilly&Associates, Inc.; 2000; ISBN: 1-56592-660-9
- [46] Olaf Kirch: *LINUX - Network Administrators Guide*, O'Reilly&Associates, Inc.; 1995; ISBN: 1-56592-087-2
- [47] Hal Stern, Mike Eisler & Ricardo Labiaga: *Managing NFS and NIS*, O'Reilly&Associates, Inc.; 2001; ISBN: 1-56592-510-6
- [48] Paul Albitz & Cricket Liu: *DNS and Bind*, O'Reilly&Associates, Inc.; 2001; ISBN: 1-596-00158-4
- [49] James M. Kretchmar: *Open Source Network Administration*, Pearson Education, Inc.; 2004; ISBN: 0-13-046210-1

- [50] Douglas R. Mauro, Kevin J. Schmidt: *Essential SNMP(1.nd Ed.,* O'Reilly&Associates, Inc.; 2001; ISBN: 0-596-00020-0
- [51] Douglas R. Mauro, Kevin J. Schmidt: *Essential SNMP(2.nd Ed.,* O'Reilly&Associates, Inc.; 2005; ISBN: 0-596-00840-6
- [52] James D. Murray: *Wimdows NT SNMP,* O'Reilly&Associates, Inc.; 1998; ISBN: 1-56592-338-3
- [53] Marshall T. Rose: *The Simple Book(2nd. Ed.),* Prentice Hall, Inc.; 1996; ISBN: 0-13-451659-1
- [54] David Perkins, Evan McGinnis: *Understanding SNMP MIBs,* Prentice Hall, Inc.; 1997; ISBN: 0-13-437708-7
- [55] Mathias Hein, David Griffiths: *SNMP Simple Network Management Protocol Version 2,* International Thomson Publishing GmbH; 1994; ISBN: 3-929821-51-6
- [56] Peter Erik Mellquist: *SNMP++ An Object-Oriented Approach to Developing Network Management Applications,* Hewlett-Packard(TM) Professional Books; 1997; ISBN: 0-13-264607-2
- [57] Marshall T. Rose: *The Open Book - A Practical Perspective on OSI,* Prentice Hall, Inc.; 1990; ISBN: 0-13-643016-3
- [58] Uyles Black: *Network Management Standards(2nd.Ed.),* McGraw-Hill, Inc.; 1994; ISBN: 0-07-005570-X
- [59] Wolfgang Barth: *NAGIOS - System and Network Monitoring,* No Starch Press, Inc.; 2006; ISBN: 1-59327-070-4
- [60] Heinz-Gerd Hegering, Sebastian Abeck, Bernhard Neumair: *Integriertes Management vernetzter Systeme,* dpunkt.verlag; 1999; ISBN: 3-932588-16-9
- [61] Walter Gora, Reinhard Speyerer: *ASN.1 Abstract Syntax Notation One(2nd.Ed.),* DATACOM-Verlag; 1990; ISBN: 3-89238-023-6
- [62] Klaus H. Stoettinger: *Das OSI-Referenzmodell,* DATACOM-Verlag; 1989; ISBN: 3-89238-021-X

Embedded Systems

- [63] Derek J. Hatley,Imtiaz A. Pirbhai: *Strategien fuer die Echtzeit-Programmierung,* Carl Hanser Verlag Muenchen-Wien; 1988; ISBN: 3-446-16288-7
- [64] Edmund Jordan: *Embedded Systeme mit Linux programmieren - GNU-Software tools zur Programmierung ARM-basierender Systeme,* Franzis Verlag GmbH; 2004; ISBN: 3-7723-5599-4
- [65] Michael Barr, Anthony Massa: *Programming Embedded Systems(2nd.Ed.),* O'Reilly&Associates, Inc.; 2006; ISBN: 1-596-00983-6
- [66] John Lombardo: *Embedded Linux,* New Riders Publishing; 2001; ISBN: 0-7357-0998-X
- [67] Craig Hollabaugh, Ph.D.: *Embedded Linux - Hardware, Software, and Interfacing,* Addison Wesley, Inc.; 2002; ISBN: 0-672-32226-9

- [68] Bob Smith, John Hardin, Graham Phillips, and Bill Pierce: *LINUX APPLIANCE DESIGN - A Hands-On Guide to Building Linux Appliances*, No Starch Press, Inc.; 2007; ISBN: 978-1-59327-140-4
- [69] David E. Simon: *An Embedded Software Primer*, Addison Wesley, Inc.; 1999; ISBN: 0-201-61569-X
- [70] John Waldron: *Introduction to RISC Assembly Language Programming*, Addison Wesley, Inc.; 1999; ISBN: 0-201-39828-1
- [71] Steve Furber: *ARM System Architecture*, Addison Wesley, Inc.; 1996; ISBN: 0-201-40352-8

Online References

Obvious, but to be written due to German-Law:

"It should be recognized, that the given links within this and following sections are solely owned by and are within the exclusive responsibility of their owners. The intention to reference to that sites is neither to take ownership of their work, nor to state commitment to any given statement on their sites. It is much more to honour the work, and thank to the help, the author advanced from by himself. Last but not least, the intention is to support a short cut for the users of UnifiedSessionsManager to sources of required help."

OSs

- [72] RedHat(TM) Enterprise Linux: <http://www.redhat.com>
- [73] RedHat(TM) Enterprise Linux-Doc: <http://www.redhat.com/docs>
- [74] CentOS: <http://www.centos.org>
- [75] CentOS-Doc: <http://www.centos.org/docs>
- [76] Scientific Linux: <http://www.scientificlinux.org>
- [77] Debian: <http://www.debian.org>
- [78] OpenSuSE: <http://www.opensuse.org>

- [79] OpenBSD: <http://www.openbsd.org>
- [80] OpenBSD-FAQ: <http://www.openbsd.org/faq/index.html>
- [81] OpenBSD-PF: <http://www.openbsd.org/faq/pf/index.html>
- [82] FreeBSD: <http://www.freebsd.org>
- [83] NetBSD: <http://www.netbsd.org>

- [84] uCLinux: <http://www.uclinux.org>

- [85] Linux on ARM: <http://www.linux-arm.com>
- [86] eCos: <http://ecos.sourceforge.org>

Hypervisors/Emulators

kvm

- [87] KVM: http://de.wikipedia.org/wiki/Kernel_based_Virtual_Machine

QEMU

- [88] QEMU(TM): <http://www.qemu.org>
- [89] QEMU-CPU support: <http://fabrice.bellard.free.fr/qemu/status.html>
- [90] QEMU-User Manual:
<http://fabrice.bellard.free.fr/qemu/qemu-doc.html>
- [91] QEMU-OS support: <http://www.claunia.com/qemu>
- [92] QEMU-Debian: <http://909ers.apl.washington.edu/dushaw/ARM>
- [93] QEMU-Debian on an emulated ARM machine; Aurelien Jarno:
<http://www.aurel32.net>
- [94] QEMU-Debian kernel and initrd for qemu-arm-versatile; Aurelien Jarno: <http://people.debian.org/aurel32>
- [95] QEMU-Debian ARM Linux on QEMU; Brian Dushaw:
<http://909ers.apl.washington.edu/dushaw/ARM>
- [96] QEMU-Ubuntu-Installation/QemuEmulator:
<http://help.ubuntu.com/community/Installation/QemuEmulator>
- [97] QEMU-Ubuntu-VMwarePlayerAndQemu:
<http://wiki.ubuntu.com/VMwarePlayerAndQemu>
- [98] QEMU-OpenBSD: <http://141.48.37.144/openbsd/qemu.html> - *Dag Leine Institut fuer Physikalische Chemie, Martin Luther Universitaet Halle*
- [99] QEMU-NetBSD - Running NetBSD on emulated hardware:
<http://www.netbsd.org/ports/emulators.html>
- [100] QEMU-OpenSolaris: <http://www.opensolaris.org/os/project/qemu/host>
- [101] QEMU-OpenSolaris Networking:
http://www.opensolaris.org/os/project/qemu/Qemu_Networking
- [102] QEMUlator: <http://qemulator.createweb.de>

SkyEye

- [103] SkyEye: <http://skyeye.sourceforge.net>

VMware

- [104] VMware(TM) Inc.: <http://www.vmware.com>
- [105] VMware-OpenBSD - HowTo Install VMWare-Tools:
<http://openbsd-wiki.org>
- [106] VMware-Communities: <http://communities.vmware.com>
- [107] VMware-Forum: <http://vmware-forum.de>
- [108] VMware-Any-Any-Patch from Petr Vandrovec:
<http://kinikovny.cvnt.cz/ftp/pub/vmware>
- [109] Open Virtual Machine Tools: <http://open-vm-tools.sourceforge.net>

Xen

- [110] RED HAT(TM) ENTERPRISE LINUX 5.1 - Virtualization:
<http://www.redhat.com/rhel/virtualization>
- [111] Virtual Machine Manager - VMM:
<http://virtual-manager.et.redhat.com>
- [112] libvirt: <http://www.libvirt.org>
- [113] Xen(TM): <http://www.xen.org>
- [114] XenWiki - Xen-Networking: <http://wiki.xensource.com/xenwiki/XenNetworking>
- [115] Xen-CentOS: *Creating and Installing a CentOS5 domU instance*
- [116] Xen-Fedora: <http://fedoraproject.org/wiki/FedoraXenQuickstartFC6>
- [117] Xen-OpenSolaris: <http://www.opensolaris.org/os/community/xen>
- [118] Gerd Hoffmann: *"Install SuSE as Xen guest."*

Security

- [119] OpenSSH: <http://www.openssh.org>
- [120] SSH Communications Security(TM): <http://www.ssh.com>
- [121] SSH Tectia(TM): *SSH Tectia*
- [122] OpenSSH-FAQ: <http://www.openbsd.org/openssh/faq.html>
- [123] OpenSSL: <http://www.openssl.org>
- [124] OpenLDAP: <http://www.openldap.org>
- [125] MIT-Kerberos: <http://web.mit.edu/kerberos/www>
- [126] Heimdal-Kerberos: <http://www.h5l.org>
- [127] sudo - "superuser do"(check google for actual link):
<http://www.sudo.ws>

Specials

Dynagen/Dynamips

- [128] Dynagen, by Greg Anuzelli:
Dynagen
- [129] Dynamips, Cisco(TM) 7200 Simulator:
Dynamips, Cisco(TM) 7200 Simulator

QEMU-Networking with VDE

- [130] VDE-Virtual Distributed Ethernet:
<http://sourceforge.net/projects/vde>
- [131] VirtualSquare: *<http://virtualsquare.org>*
- [132] VirtualSquare: *Basic Networking*

PXE

- [133] By H. Peter Anvin: *SYSLINUX - PXELINUX - ISOLINUX*
- [134] PXE-ROM-Images: *Etherboot*

Routing

- [135] "Linux Advanced Routing&Traffic Control", by Bert Hubert:
LARTC-HOWTO

Scratchbox

- [136] *Scratchbox*

Serial-Console

- [137] By van Emery: *"Linux Serial Console HOWTO"*
- [138] By David S. Lawyer / Greg Hawkins: *"Serial-HOWTO"*
- [139] By David S. Lawyer: *"Text-Terminal-HOWTO"*

Miscellaneous

- [140] IEEE: *<http://www.ieee.org>*
- [141] The GNU Netcat: *Netcat*
- [142] Netcat Wikipedia: *Netcat Wikipedia*
- [143] By van Emery: *"Linux Gouge"*

UnifiedSessionsManager Versions

- [144] The first public version of 2008.02.11, by Arno-Can Uestuensoez. Available with online ASC-II help only by "ctys -H print": "*UnifiedSessionsManager*" <http://sourceforge.net/projects/ctys>
- [145] The second public version of 2008.07.10, by Arno-Can Uestuensoez: "*UnifiedSessionsManager*" <http://sourceforge.net/projects/ctys>
- [146] The second public version with minor updates of 2008.08.6, by Arno-Can Uestuensoez: "*UnifiedSessionsManager*" <http://sourceforge.net/projects/ctys>
- [147] Minor editorial updates of 2008.08.12, by Arno-Can Uestuensoez: "*UnifiedSessionsManager*" <http://sourceforge.net/projects/ctys>
- [148] Enhanced documentation, 2008.08.16, by Arno-Can Uestuensoez: "*UnifiedSessionsManager*" <http://sourceforge.net/projects/ctys>
- [149] Minor editorial changes, 2009.01.26, by Arno-Can Uestuensoez: "*UnifiedSessionsManager*" <http://sourceforge.net/projects/ctys>

Sponsored OpenSource Projects

- [150] UnifiedSessionsManager:
<http://www.UnifiedSessionsManager.org>
<http://sourceforge.net/projects/ctys>
<http://ctys.berlios.de>
- [151] zamklibant:
<http://sourceforge.net/projects/zamklibant>

Commercial Support

Commercial support and additional services are available exclusively from:

- [152] Arno-Can Uestuensoez:
<http://www.uestuensoez.com>
- [153] Ingenieurbuero Arno-Can Uestuensoez:
<http://www.i4p.com>